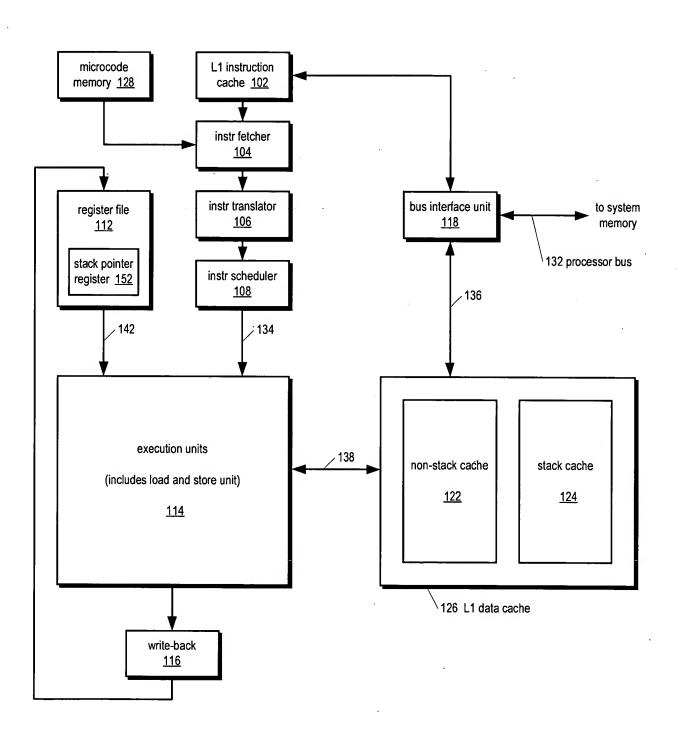
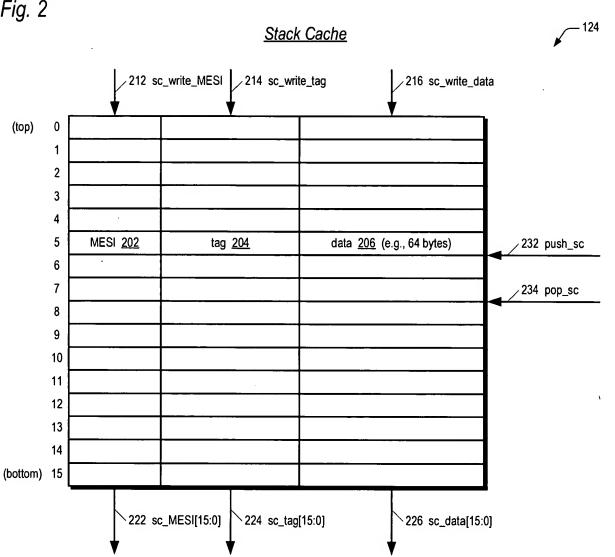


Fig. 1

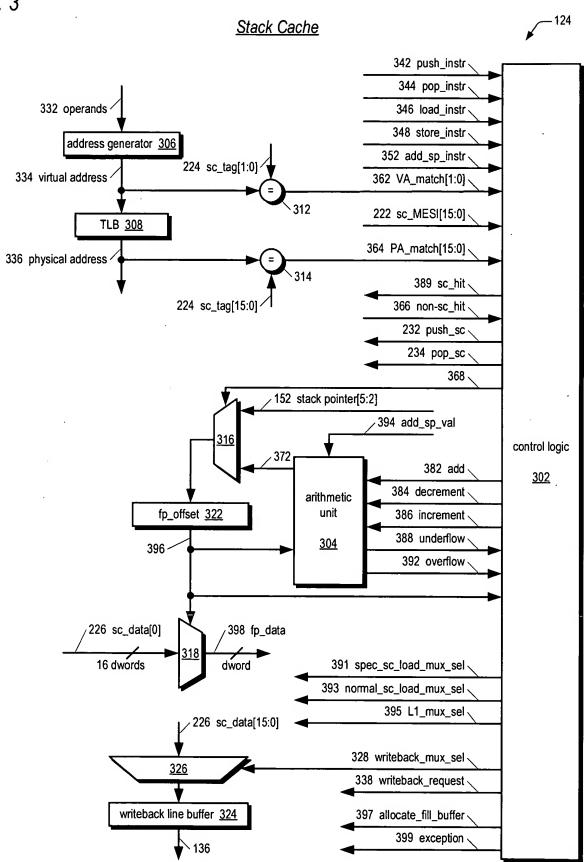


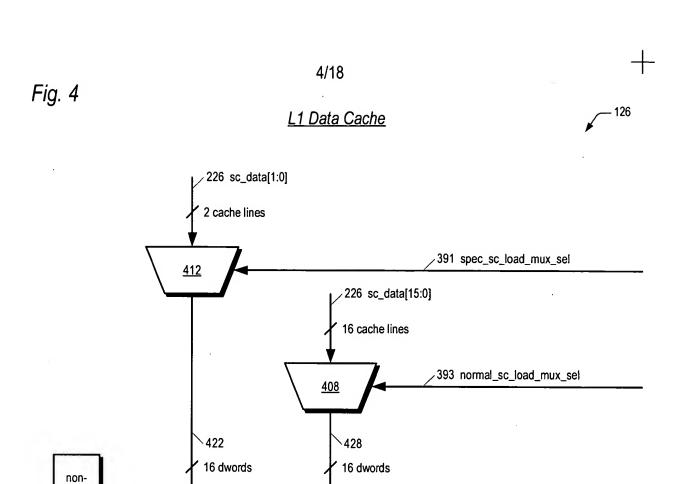












<u>406</u>

\426

336 physical address[5:2]

398 fp_data

395 L1_mux_sel

stack cache

<u>122</u>

432

<u>404</u>

^424

<u>402</u>

138

Fig. 5

Fast Pop Operation

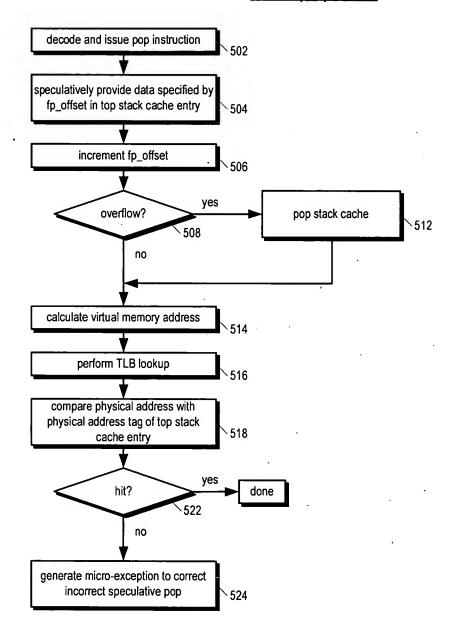
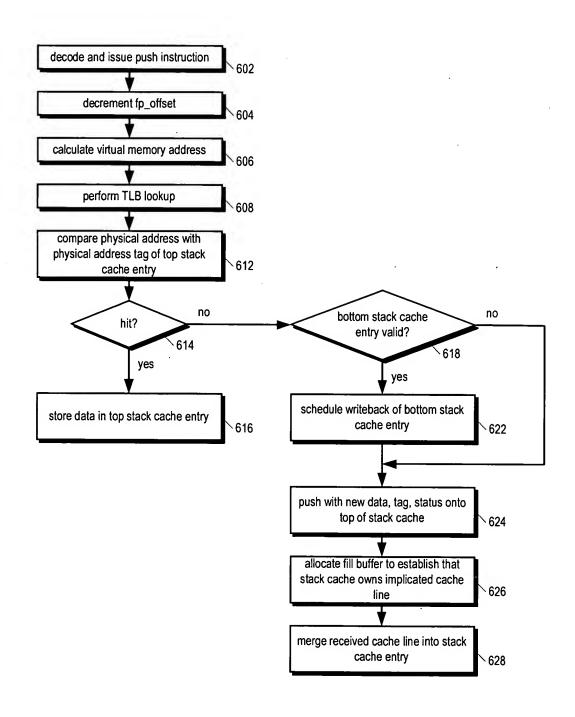


Fig. 6

Push Operation



+-

Fig. 7

Add to Stack Pointer Operation

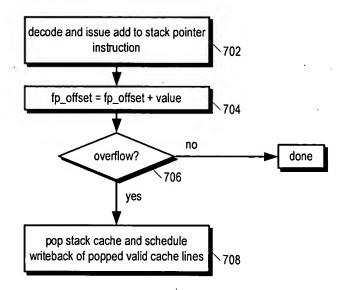


Fig. 8A

Speculative Load from Stack Cache Operation

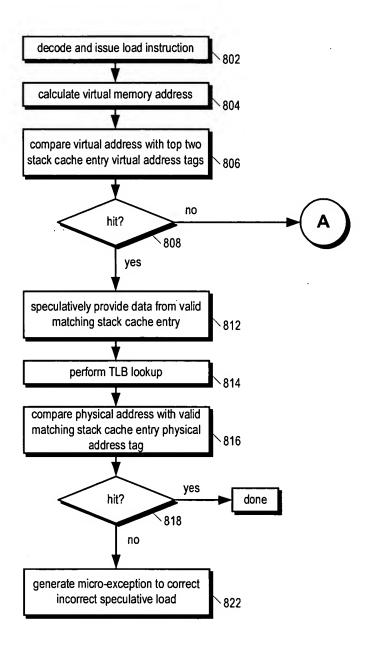


Fig. 8B

Normal Load from Stack Cache Operation

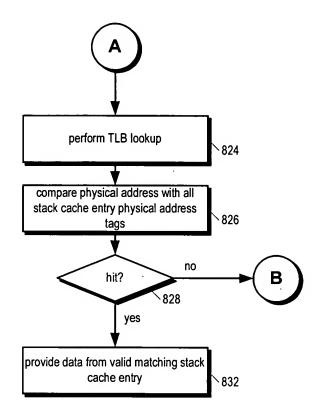


Fig. 8C

Load from Non-Stack Cache Operation

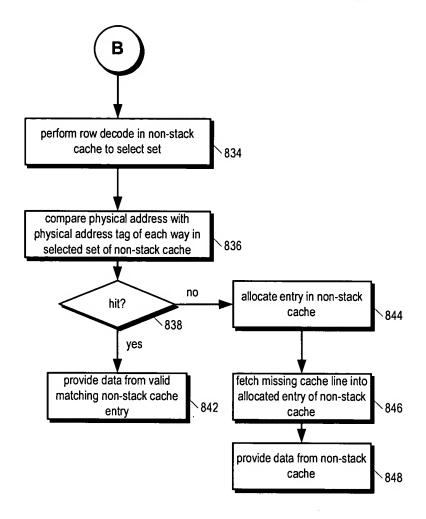
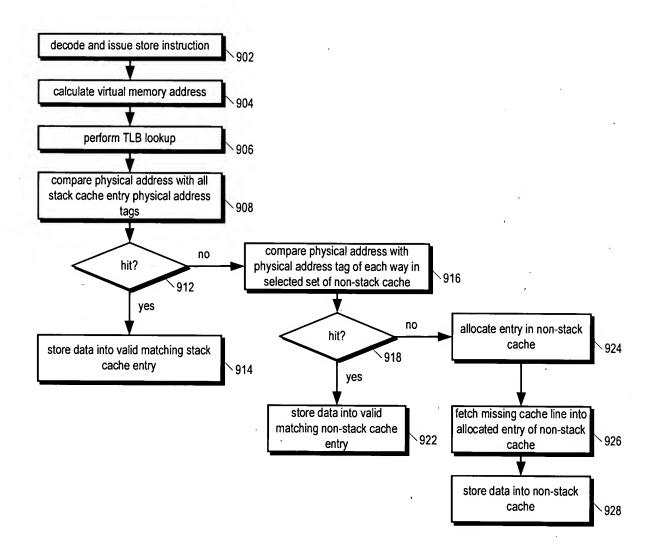


Fig. 9

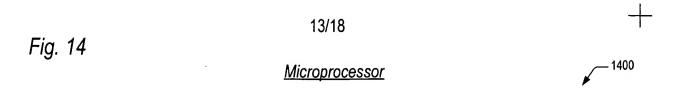
Store Operation

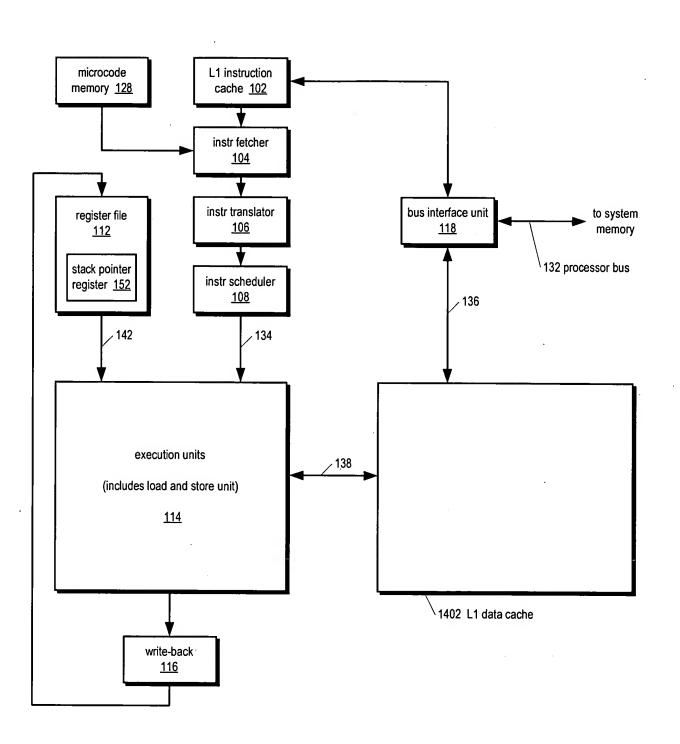


 \dotplus

12/18 Fig. 10 Fast Pop from Stack Cache Timing clock cycle ==> 1 2 4 receive pop instruction request pop mux dword from cache line in top entry based on fp_offset pop calculate virtual address pop perform TLB lookup pop detect incorrect stack cache pop, based on physical address compare pop data available Fig. 11 Speculative Load from Stack Cache Timing 1 2 clock cycle ==> 3 4 receive load instruction request load calculate virtual address load virtual tag compare and generate speculative load select from matches and valids load perform TLB lookup load mux cache line based on speculative load select and mux dword based on PA[5:2] load detect incorrect speculative load, based on physical address compare load data available Fig. 12 Normal Load from Stack Cache Timing clock cycle ==> 1 2 3 5 receive load instruction request load calculate virtual address load perform TLB lookup load physical tag compare and generate normal load select from matches and valids load mux cache line based on normal load select and mux dword based on PA[5:2] load data available Fig. 13 Load from Non-Stack Cache Timing clock cycle ==> 1 2 3 5 6 receive load instruction request load calculate virtual address load perform TLB lookup load row decode based on physical address index and array lookup load physical tag compare and generate way select based on matches and valids load mux cache line based on way select and mux dword based on PA[5:2] load

data available





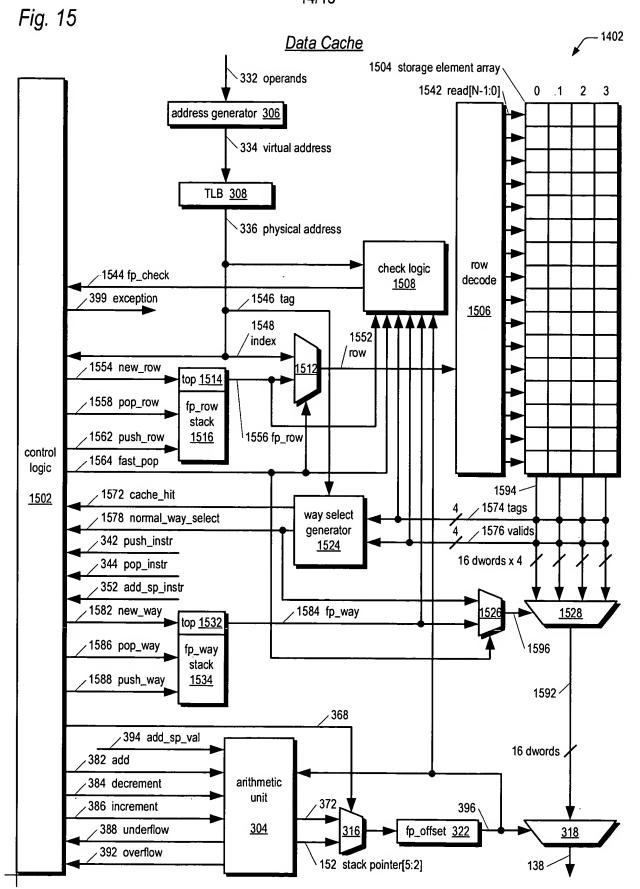


Fig. 16

Fast Pop Operation

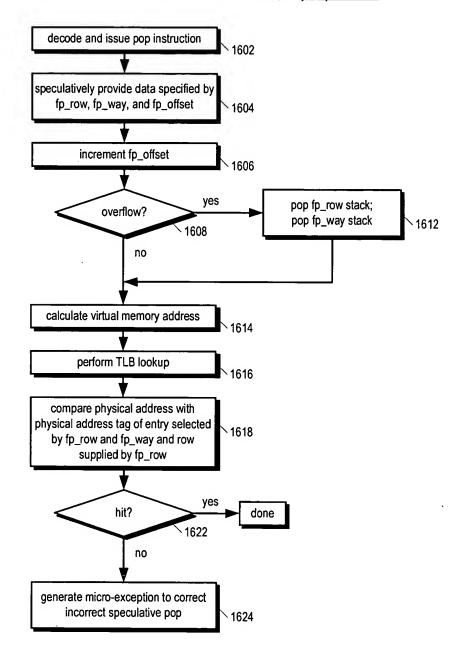


Fig. 17

Push Operation

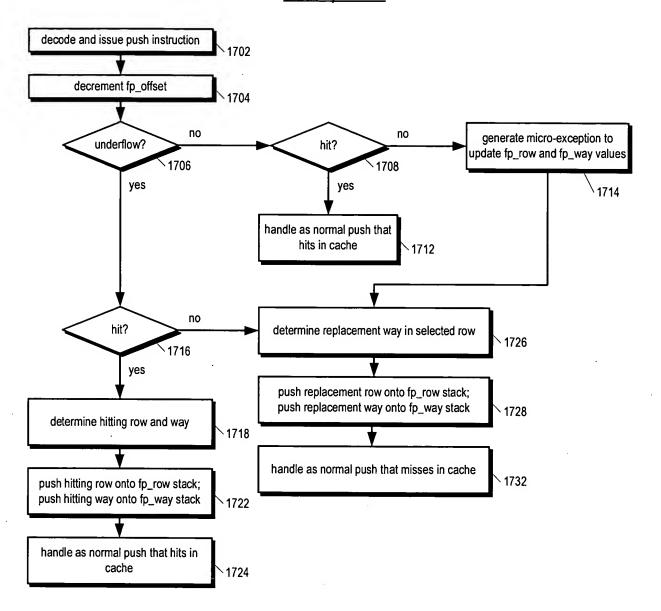


Fig. 18

Add to Stack Pointer Operation

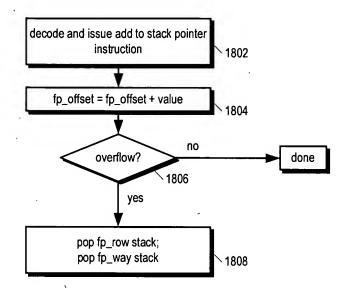


Fig. 19

18/18 Fast Pop from Cache Timing

·				
clock cycle ==>	1	2	3	4
receive pop instruction request	рор			
row decode based on fp_row and array lookup		pop		
calculate virtual address		pop		
mux cache line based on fp_way and mux dword based on fp_offset			pop	
perform TLB lookup			pop	
detect incorrect speculative pop, based on physical address compare				рор
data avai	lable—			